



Docket No.: 826.1654

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Yoshiyuki ITO

Serial No. 09/748,370

Group Art Unit: 2165

Confirmation No. 3850

Filed: December 27, 2000

Examiner: Neveen Abel Jalil

For: INFORMATION EXTRACTION DEVICE AND STORAGE MEDIUM

**SUBMISSION OF VERIFIED ENGLISH TRANSLATION**

Commissioner for Patents  
PO Box 1450  
Alexandria, VA 22313-1450

Sir:

Attached is the English translation of Japanese Patent Application No. 2000-21496, filed January 31, 2000 which, as stated in the Verification signed by the translator, has been claimed for priority in the subject U.S. patent application. It is respectfully requested that the attached English translation be made of record in the above-identified application.

If any additional fees are required in connection with the filing of this document, please charge Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 8/12/05

By: Richard A. Gollhofer  
Richard A. Gollhofer  
Registration No. 31,106

1201 New York Ave, N.W., Suite 700  
Washington, D.C. 20005  
Telephone: (202) 434-1500  
Facsimile: (202) 434-1501

VERIFICATION

I, Hanae SASADA, residing at Hyogo, Japan, state:  
that I know well both the Japanese and English languages;  
that I translated, from Japanese into English, the  
priority document as filed in the U.S. Patent  
Application No. 09/748,370, filed on December 27, 2000;  
and that the attached English translation is a true  
and accurate translation to the best of my knowledge  
and belief.

Dated: July 22, 2005

Hanae Sasada.

Hanae SASADA

Reference No. 99-52050

[Document Name]	Patent Application
[Reference No.]	9952050
[Filing Date]	January 31, 2000
[Addressee]	Commissioner, Patent Office
[Int'l Patent Classification]	G06F 15/00
[Name of the Invention]	Information Extraction Device and Storage Medium
[Number of Claims]	4
[Inventor]	
[Address or Residence]	c/o FUJITSU LIMITED, 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi, Kanagawa
[Name]	Yoshiyuki Ito
[Patent Applicant]	
[Identifying No.]	000005223
[Name]	FUJITSU LIMITED
[Agent]	
[Identifying Number]	100089141
[Address or Residence]	21-20-603, Hiramachi 1-chome, Meguro-ku, Tokyo
[Attorney]	
[Name or Title]	Morihiro OKADA
[Telephone No.]	03-3725-2215
[Fee Designation]	
[Pre-payment Reg. No.]	015543
[Payment Amount]	JPY21,000
[Index of Submitted Article]	
[Article Name]	Specification 1
[Article Name]	Drawings 1
[Article Name]	Abstract 1
[General Power of Attorney No.]	9705795
[Necessity of Proof]	Yes

[Document Name] Specification

[Title of the Invention] INFORMATION

EXTRACTION DEVICE AND STORAGE MEDIUM

[What is Claimed is:]

- 5 1. An information extraction device for extracting information corresponding to a request from a database, comprising:
- a function called up from application software, for converting a plurality of kinds of
- 10 requests into requests in an XML format by referring to a database and returning results.
2. The information extraction device according to claim 1, wherein a list of requests related to the generated requests in an XML format that
- 15 can be arbitrarily selected and set, is linked and provided.
3. The information extraction device according to claim 1 or 2, wherein a list of requests related to information designated in the generated
- 20 requests in an XML format that can be arbitrarily selected and set, is linked and provided.
4. A computer-readable storage medium on which is recorded a program for extracting information corresponding to a request from a database as an
- 25 information extraction device and converting a

plurality of kinds of requests into requests in an XML format by referring to a database as a function called up from application software.

[Detailed Explanation of the Invention]

5 [0001]

[Field of the Invention]

The present invention relates to an information extraction device for converting a plurality of kinds of requests into requests in an XML format by referring to a database, extracting information from the requests and replying to the requests and a storage medium thereof.

[0002]

15 [Prior Art Technology]

Conventionally, when extracting a plurality of pieces of information from a large-sized database managed by a server via a network, such as the Internet, a user (user terminal) set the extraction of a plurality of pieces of information in an object and transmitted the object to the server, and the server set a plurality of pieces of information extracted from the database in the object and returned the object. Then, the user extracted the plurality of pieces of information using an

interface prepared for the object.

[0003]

[Problems to be Solved by the Invention]

However, according to the conventional  
5 method described above, a range from which  
information can be extracted, the kind of  
information which can be extracted by the interface  
of an object, etc., are restricted, which is a  
problem.

10 [0004]

It is an object of the present invention to  
easily and rapidly extract information about a  
plurality of kinds of requests and to return the  
information by a server receiving requests from  
15 a user and by a function called up from the  
application software of the server converting the  
plurality of kinds of requests into requests in  
an XML format by referring to a database, by  
extracting results from the requests and returning  
20 the results to the user to solve these problems.

[0005]

[Means for Solving the Problems]

Means for solving the problems is described  
with reference to Fig. 1.

25 In Fig. 1, a server 2 is connected to a network

and provides a variety of services. In this example, the server 2 comprises an application software 3, an XML generation function 4, etc.  
[0006]

5           Application software 3 executes a variety of processes according to a program. In this example, the application software 3 receives requests from a browser 1, transfers the received requests to the XML generation function 4, makes  
10   the XML generation function generate requests in an XML format, etc.  
[0007]

          The XML generation function 4 converts the plurality of kinds of requests transferred from  
15   the application software 3 into requests in an XML format, etc.

          A database 6 stores a variety of information in order to simplify the retrieval of information.  
[0008]

20           Next, the operation is described.

          The application software 3 of the server 2 transfers requests received from the browser 1 to the XML generation function 4. The XML generation function 4 converts the transferred  
25   plurality of kinds of requests into requests in

an XML format by referring to the database 6 and returns the requests to the application software 3. The application software 3 extracts information from the requests and replies to the browser 1.

[0009]

At this time, a list of requests related to the generated requests in an XML format that can be arbitrarily selected or set is linked and provided.

Alternatively, a list of requests related to information designated in generated requests in an XML format that can be arbitrarily selected or set is linked and provided.

[0010]

Therefore, since the server 2 receives requests from the browser 1, a function called up from the application software 3 of the server 2 converts the plurality of kinds of requests into requests in an XML format by referring to the database 6, extracting results from the requests and returning the results to the browser 1, information about a plurality of kinds of requests can be easily and rapidly extracted and can be returned to the browser 1.



[0011]

[Preferred Embodiments]

The preferred embodiments of the present invention and the operation thereof are described in detail with reference to Figs. 1 through 15.

[0012]

Fig. 1 shows the system configuration of the present invention.

In Fig. 1, the browser 1 is used for a user to transmit requests to the server 2 via a network, such as the Internet, etc., to display the returned retrieval results of the database 6, etc.

[0013]

The server 2 loads a program read from a storage medium, which is not shown in Fig. 1, into a main memory, starts the program to execute a variety of processes described below. In this example, the server 2 comprises the application software 3, an XML generation function 4, an XML operation function 5, etc.

[0014]

The application software 3 is a program. In this case, the application software 3 receives a request from the browser 1, transfers the received request to the XML generation function

4, makes the XML generation function generate a request in an XML format by referring to the database 6, extracts information from the request, returns the extracted information to the browser 1, and the like.

[0015]

The XML generation function 4 converts the plurality of kinds of requests transferred from the application software 2 into requests in an XML format by referring to the database 6, and the like.

[0016]

The XML operation function 5 performs a variety of operations, such as extracting information from the database 6 according to the request in an XML format, etc.

The database 6 stores a variety of information shown later in Fig. 7 for the purpose of easy retrieval.

20 [0017]

Next, the operation of the configuration shown in Fig. 1 is described in detail with reference to Figs. 2 through 5.

Fig. 2 is a flowchart showing the operation of the present invention.

In Fig. 2, in S0, REQ is referenced. In this case, the REQ part of a request transmitted by the browser 1 shown in Fig.1 via a network is referenced. For example, REQ (request) of the underlined part after ? (Example 1 shown later in Fig. 6) in the following address (address, including a request) is extracted and referenced.

http://www. (address of a home page)  
 /cgi.exe?REQ=GET PROP&REQ TYPE=DOCUMENT&DOC  
 10 ID=286

In this case, it is judged which of the followings is REQ referenced in S0.

-GET PROP (in the case of Example 1 shown in Fig. 6)

15 -BROWSE (in the case of Example 2 shown in Fig. 6)

-GET REL (in the case of Example 3 shown in Fig. 6)

[0018]

20 In the case of S1 where in S0, REQ (request) is referenced and GET PROP is detected, S2 and after are executed.

In S2, REQ TYPE is referenced. In this case, since REQ TYPE=DOCUMENT is described in REQ (request) of Example 1 shown in Fig. 6 and DOCUMENT  
 25

is detected, the flow proceeds to S3. In other cases, corresponding processes are executed in the same way.

[0019]

5           In S3, DOC ID is referenced. In this case, DOC ID=286 is described in REQ (request) of Example 1 shown in Fig. 6, 286 is detected.

          In S4, property is inserted (only a designated document portion). In this process,  
10       the following process of A (A0) shown in Fig. 3 is executed.

[0020]

          In S31 shown in Fig. 3, a process is started.

          In S32, a structure tag is generated. In  
15       this case, for example, a structure tag is generated, as shown in part AT1 of the request (Example 1) in an XML format shown in Fig. 8 described later.

[0021]

20           In S33, column information required by a user is extracted from a database. In this case, for example, the GP-A0 of the request in an XML format shown later in Fig. 8 is extracted from the database and is inserted.

25       [0022]

In S34, a tag is generated. In this case, for example, a tag shown in part AT2 of the request in an XML format shown later in Fig. 8 is generated.

In S35, the process is terminated. Then,  
5 in this case, the flow returns to S4 shown in Fig. 2 and proceeds to S5.

[0023]

In S5, a process for inserting property definition is executed. In this process, the  
10 following process (B0) for inserting the information about property shown later in Fig. 4 is executed.

In S41 shown in Fig. 4 a process is started.

[0024]

15 In S42, information about property definition is obtained from the database 6. In this case, for example, part GP-B0 of the request in an XML format shown later in Fig. 8 is extracted from the database and is inserted.

20 [0025]

In S43, a structure tag is generated. In this case, for example, a structure tag shown in part BT1 of the request in an XML format shown later in Fig. 8 is generated.

25 In S44, a tag is generated. In this case,

for example, a tag shown in part BT2 of the request in an XML format shown later in Fig. 8 is generated.

[0026]

In S45, the process is terminated. Then,  
5 in this case, the flow returns to S5 shown in Fig. 2 and is terminated.

In S0 described above, if REQ (request) is, for example, Example 1 shown in Fig. 6, the REQ is referenced, GET PROP is detected, and in S1  
10 through S5 the request (Example 1) in an XML format shown later in Fig. 8 can be automatically generated (converted).

[0027]

Similarly, in the case of S11 shown in Fig. 2 where in S0 REQ (request) is referenced and BROWSE  
15 is detected, S12 and after are executed.

In S12, MY CONT ID is referenced. In this case, MY CONT ID=22 is described in REQ (request) in Example 2 shown in Fig. 6 and 22 is detected.

20 [0028]

In S13, property is inserted (in a designated container). In this process, the following process of A (A1) shown in Fig. 3 is executed.

In S31 shown in Fig. 3, a process is started.

25 [0029]

In S32, a structure tag is generated. In this case, for example, a structure tag is generated, as shown in part AT1 of the request (Example 2) in an XML format shown later in Figs.

5 10 and 11.

[0030]

In S33, column information required by a user is extracted from the database. In this case, for example, part BR-A1 of the request in an XML format shown later in Fig. 10 is extracted from the database and is inserted.

[0031]

In S34, a tag is generated. In this case, for example, a tag shown in part AT2 of the request in an XML format shown later in Figs. 10 and 11 is generated.

In S35, the process is terminated. Then, the flow returns to S13 and proceeds to S14.

[0032]

20 In S14, correlation is inserted. In this process, the following process of C (C0) shown in Fig. 5 is executed.

In S51 shown in Fig. 5, a process is started.

[0033]

25 In S52 a structure tag required to indicate

correlation is generated. In this case, for example, a structure tag shown in part CT1 of the request (Example 2) in an XML format shown later in Figs. 10 and 11 is generated.

5 [0034]

In S53, information about correlation is extracted. In this case, information about correlation is extracted from the database 6 and, for example, parts BR-CO1 and BR-CO2 of the request  
10 in an XML format shown later in Figs. 10 and 11 are inserted.

[0035]

In S54, correlation information is inserted. In this case, for example, part CT2 of the request  
15 in an XML format shown later in Figs. 10 and 11 is inserted.

In S55, the process is terminated. Then, the flow returns to S14 and proceeds to S15.

[0036]

20 In S15, property is inserted (in a designated container). In this process, the process of A (A2) shown in Fig. 3 is executed in the same way.

In S16, property definition is inserted. In this process, the process of B (B1) shown in Fig.  
25 4 is executed in the same way.



[0037]

In S0 described above, if REQ (request) is, for example, Example 2 shown in Fig. 6, the REQ is referenced, BROWSE is detected, in S11 through  
5 S16 the request (Example 2) in an XML format shown later in Figs. 10 and 11 can be automatically generated (converted).

[0038]

In the case of S21 shown in Fig. 2 where in  
10 S0 REQ (request) is referenced and GET REL is detected, S22 and after are executed.

In S22, TYPE is referenced. In this case, since TYPE=DESTINATION is described in REQ (request) of Example 3 shown in Fig. 6 and  
15 DESTINATION is detected, the flow proceeds to S23. In other cases, respective corresponding processes are executed in the same way.

[0039]

In S23, DOC ID is referenced. In this case,  
20 DOC ID=100 is described in REQ of Example 3 shown in Fig. 6, and 100 is detected.

In S24, property is inserted (in a designated document). In this process, the following process of A (A3) shown in Fig. 3 is executed.

25 [0040]

In S31 shown in Fig. 3, a process is started.

In S32, a structure tag is generated. In this case, for example, a structure tag shown in part AT1 of the request (Example 3) in an XML format  
5 shown later in Figs. 13 and 14 is generated.  
[0041]

In S33, column information required by a user is extracted from the database. In this case, for example, part GR-A3 of the request in an XML format  
10 shown later in Figs. 13 and 14 is extracted from the database and is inserted.  
[0042]

In S34, a tag is generated. In this case, for example, a tag shown in part AT2 of request  
15 in an XML format shown later in Figs. 13 and 14 is generated.

In S35, the process is terminated. Then, the flow returns to S24 shown in Fig. 2 and proceeds to S25.  
20 [0043]

In S25, correlation is inserted. In this process, the following process of C (C1) shown in Fig. 5 is executed.

In S51 shown in Fig. 5, a process is started.  
25 [0044]

In S52, a structure tag required to indicate correlation is generated. In this case, for example, a structure tag shown in part CT1 of the request (Example 3) in an XML format shown later  
5 in Figs. 13 and 14 is generated.

[0045]

In S53, information about correlation is extracted. In this case, information about correlation is extracted from the database 6, and,  
10 for example, part GR-C11 of the request in an XML format shown later in Figs. 13 and 14 is inserted.

[0046]

In S54, correlation information is inserted. In this case, for example, CT2 of an XML format  
15 shown later in Figs. 13 and 14 is inserted.

In S55, the process is terminated. Then, in this case, the flow returns to S25 shown in Fig. 2 and proceeds to S26.

[0047]

20 In S26, property is inserted (in an extracted document). In this process, the process of A (A4) shown in Fig. 3 is executed in the same way.

In S27, property definition is inserted. In this process, the process of B (B2) shown in Fig.  
25 4 is executed in the same way.

[0048]

In S0 described above, if REQ (request) is, for example, Example 3 shown in Fig. 6, the REQ is referenced, GET REL is detected, and in S23  
5 through S27 the request (Example 3) in an XML format shown later in Figs. 13 and 14 can be automatically generated (converted).

[0049]

Fig. 3 is a flowchart showing the detailed  
10 operation of the present invention (process for inserting property, A). As already described with reference to Fig. 2, this flowchart describes the detailed process for inserting the property (property of a designated document, a designated  
15 container, an extracted document, etc.) of a request (REQ) in the request in an XML format shown in Figs. 8, 11, 12, 13 and 14, in the flowchart shown in Fig. 2 (for detail, see the description on S31 through S35 shown in Fig. 2).

20 [0050]

Fig. 4 is a flowchart showing the detailed operation of the present invention (information about property, B). As already described in Fig. 2, this flowchart describes the detailed process  
25 for inserting the property (property definition)

of a request (REQ) in the request in an XML format shown in Figs. 8, 11, 12, 13 and 14, in the flowchart shown in Fig. 2 (for detail, see the description on S41 through S45 shown in Fig. 2).

5 [0051]

Fig. 5 is a flowchart showing the detailed operation of the present invention (a method for inserting correlation, C). As already described in Fig. 2, this flowchart describes the detailed process for inserting the correlation of a request (REQ) in the request in an XML format shown in Figs. 8, 11, 12, 13 and 14, in the flowchart shown in Fig. 2 (for detail, see the description on S51 through S55 shown in Fig. 2).

15 [0052]

Fig. 6 shows examples of the requests of the present invention. These are examples of REQs (requests) in S1, S11 and S21 shown in already described Figs. 2 as follows.

20           Example           1:           REQ=GET           PROP&REQ  
TYPE=DOCUMENT&DOC ID=286

Example 2: REQ=BROWSE&MY CONT ID=22

Example                           3:           REQ=GET  
REL&TYPE=DESTINATION&DOC ID=100

25           These REQs (requests) are transmitted from

the browser (client) 1 to the server 2 via a network, and they are described after (the home address of the related server 2)/cgi.exe?. The details are described below using Example 1.

5 [0053]

(1) The server 2 receives http://www.(home page address)/cgi.exe?REQ=GET PROP&REQ TYPE=DOCUMENT&DOC ID=286

10 (2) Request in the underlined part of (1) is inserted in void main(int argc, char\* argv[], char\* envp[]//envp as a part of an environmental variable.

[0054]

15 (3) Request of (2) (request in the underlined part of (1)) from a user is inserted in MWISimpleCall MSCobj(envp);//envp (request extracted from the environmental variable of (2) is inserted using MWISimpleCall class).

20 [0055]

(4) A request in an XML format to be returned (see Figs. 8, 11, 12, 13 and 14) is generated (converted) and inserted in MMIRquest(MSCobj.toMWQuery(),returne  
25 dXml);//returnedXml.

[0056]

(5) Then, information is extracted from  
retunedXml of (4), is converted into an  
HTML format and is returned to the user  
5 (browser 1).

Fig. 7 shows an example of the database used  
in the present invention.

[0057]

Fig. 7(a) shows an example of a  
10 DOCUMENT(database) database. Here, the  
following information shown in Fig. 7(a) is related  
and registered.

-Document ID:  
-Name:  
15 -Writer:  
-Written day and time:  
-Document size:

Fig. 7(b) shows an example of a CONTAINER  
(container) database. Here, the following  
20 information shown in Fig. 7(b) is related and  
registered.

[0058]

-Container ID:  
-Name:  
25 Fig. 7(c) shows an example of a REFERENCE1

(reference 1) database (indicating correlation between a container and a document). Here, the following information shown in Fig. 7(c) is related and registered.

5 [0059]

-Container ID:

-Document ID:

Fig. 7(d) shows an example of a REFERENCE2 (reference 2) database (indicating correlation  
10 between containers). Here, the following information shown in Fig. 7(d) is related and registered.

[0060]

-Parent container ID:

15 -Child container ID:

Fig. 7(e) shows an example of a RELATIONSHIP (relationship) database (indicating correlation between documents). Here, the following information shown in Fig. 7(e) is related and  
20 registered.

[0061]

-Parent document ID:

-Child document ID:

-Comment:

25 -Strength:



-Inclusion:

Fig. 7(f) shows an example of a PROPERTYDEF database. The following information shown in Fig. 7(f) is related and registered.

5 [0062]

-Object ID:

-Representative Name:

-Type:

-Read flag:

10 -Necessity:

-Maximum value:

-Minimum value

Fig. 8 shows an example of the XML (Example 1) of the present invention. This request in an XML format is converted from REQ (request) of Example 1 shown in already described Fig. 6 according to the flowcharts shown in Figs. 2 through 5. Symbols used in Fig. 8 (AT1, AT2, BT1, GP-B0, CT2, GP-A0, etc.) are generated in the respective positions described in the flowcharts shown in Figs. 2 through 5 and they indicate the respective correspondences.

[0063]

(4) shown in Fig. 8 is link information. A list of requests (list of request templates)

related to document ID=286 is generated in advance and is set there. A request can be arbitrarily selected/designated and set there.

[0064]

5           Similarly, (5) shown in Fig. 8 is also link information. A list of requests (list of request templates) related to the entire or highest-order document is generated in advance and is set there. A request can be arbitrarily selected/designated  
10 and set there.

[0065]

Fig. 9 shows an example of the database (Example 1) used in the present invention. Symbols described in Fig. 9 (GP-A0, GP-B0) indicate  
15 information extracted from the respective parts corresponding to the symbols.

Figs. 10 and 11 show an example of XML (Example 2) of the present invention. This request in an XML format is converted from REQ (request) of  
20 Example 2 shown in already described Fig. 6 according to the flowcharts shown in Figs. 2 through 5. Symbols described in Figs. 10 and 11 (AT1, AT2, BR-A1, BR-A2, etc.) are generated in the respective positions described in the  
25 flowchart and indicate the respective

correspondences.

[0066]

Fig. 12 shows an example of the database (Example 2) used in the present invention. Symbols described in Fig. 12 (BR-A2, GP-B1, BR-C0, etc.) indicate respective pieces of information extracted from the respective parts corresponding to the symbols shown in Figs. 10 and 11.

[0067]

10 Figs. 13 and 14 show an example of XML (Example 3) of the present invention. This request in an XML format is converted from REQ of Example 3 shown in already described Fig. 6 according to the flowcharts shown in Figs. 2 through 5. Symbols described in Figs. 13 and 14 (AT1, AT2, GR-A3, GR-A4, etc.) are generated in the respective positions described in the flowcharts shown in Figs. 2 through 5 and they indicate the respective correspondences.

20 [0068]

Fig. 15 shows an example of the database (Example 3) used in the present invention. Symbols described in Fig. 15 (BR-A3, GR-A4, GR-C1, GR-B2, etc.) indicate respective pieces of information extracted from the respective parts

25

corresponding to the symbols shown in Figs. 13 and 14.

[0069]

[Effect of the Invention]

5           In this way, according to the present invention, since the server 2 receives requests from the browser 1, a function called up from the application software 3 of the server 2 converts the plurality of kinds of requests into requests  
10 in an XML format by referring to the database 6, transfers the request to the application software 3, the application software 3 extracts results and returns the results to the browser 1, requests in an XML format can be generated from a plurality  
15 of kinds of requests and a variety of desired information can be easily and rapidly returned to a user. If application software for publicizing the information of the database 6 to users is generated, work efficiency can be improved  
20 using an interface most familiar to an engineer generating the application software.

[Brief Descriptions of the Drawings]

Fig. 1 shows the system configuration of the present invention.

25           Fig. 2 is a flowchart showing the operation

of the present invention.

Fig. 3 is a flowchart showing the detailed operation of the present invention (a process for inserting property).

5        Fig. 4 is a flowchart showing the detailed operation of the present invention (information about property).

Fig. 5 is a flowchart showing the detailed operation of the present invention (a method for  
10 inserting correlation).

Fig. 6 shows examples of the requests of the present invention.

Fig. 7 shows an example of the database used in the present invention.

15        Fig. 8 shows an example of the XML of the present invention (Example 1).

Fig. 9 shows an example of the database used in the present invention (Example 1).

Fig. 10 shows an example of the XML of the  
20 present invention (Example 2).

Fig. 11 shows an example of the XML of the present invention (Example 2).

Fig. 12 shows an example of the database used in the present invention (Example 2).

25        Fig. 13 shows an example of the XML of the

present invention (Example 3).

Fig. 14 shows an example of the XML of the present invention (Example 3).

Fig. 15 shows an example of the database used  
5 in the present invention (Example 3).

[Explanation of the Codes]

1: Browser

2: Server

3: Application software

10 4: XML generation function

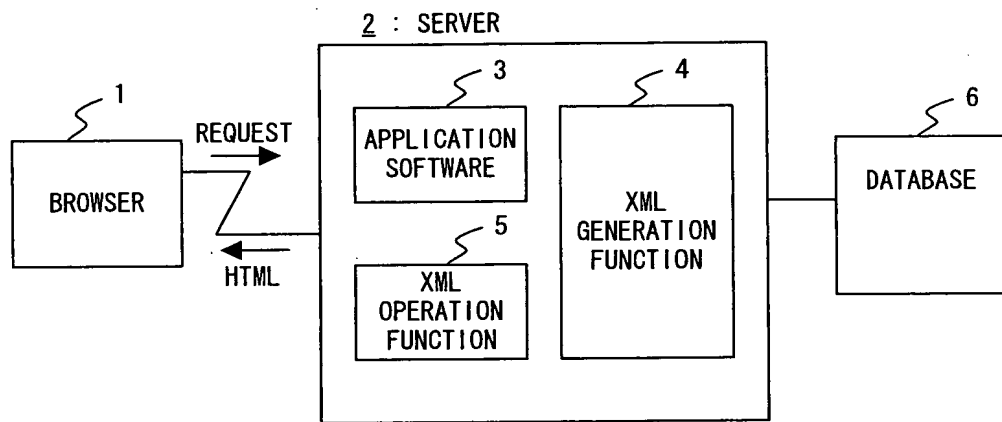
5: XML operation function

6: Database

[Document Name] Drawings

[Fig. 1]

Diagram showing the system configuration  
of the present invention



[Document Name] Abstract

[Abstract]

[Object] The present invention relates to an information extraction device for converting a plurality of kinds of requests into requests in an XML format by referring to a database and extracting information from the requests and a storage medium thereof. It is an object of the present invention to easily and rapidly extract information about the plurality of kinds of requests and to return the information to a user by a server receiving requests from the user and by a function called up from the application of the server converting the plurality of kinds of requests into requests in an XML format by referring to the database, extracting results from the request and notifying the user of the results.

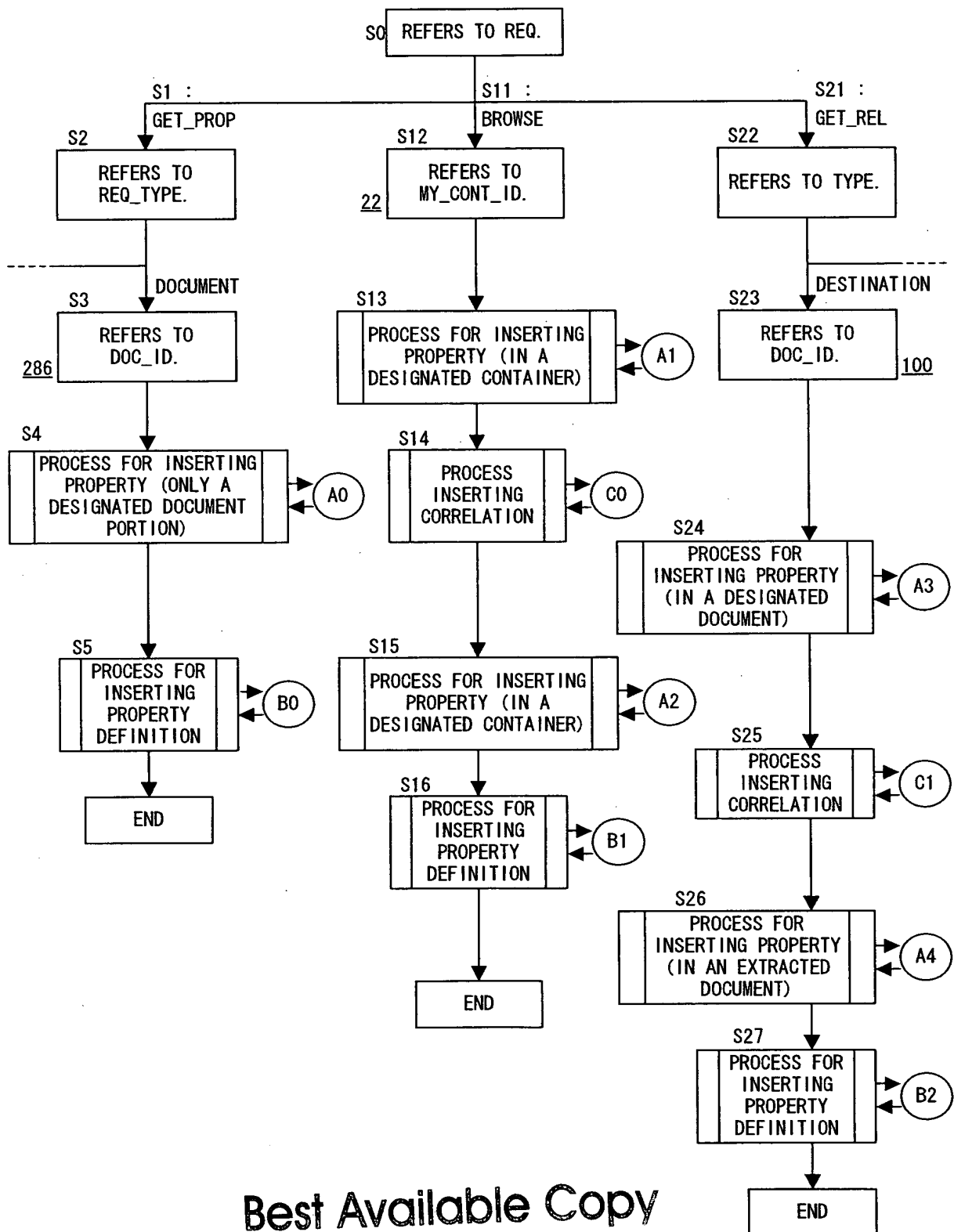
[Means for Solving the Problems] An information extraction device according to the present invention is configured so as to comprise a function called up from an application software, for converting a plurality of kinds of requests into requests in an XML format by referring to a database and returning results.

[Selected Drawing] Fig. 1



[Fig. 2]

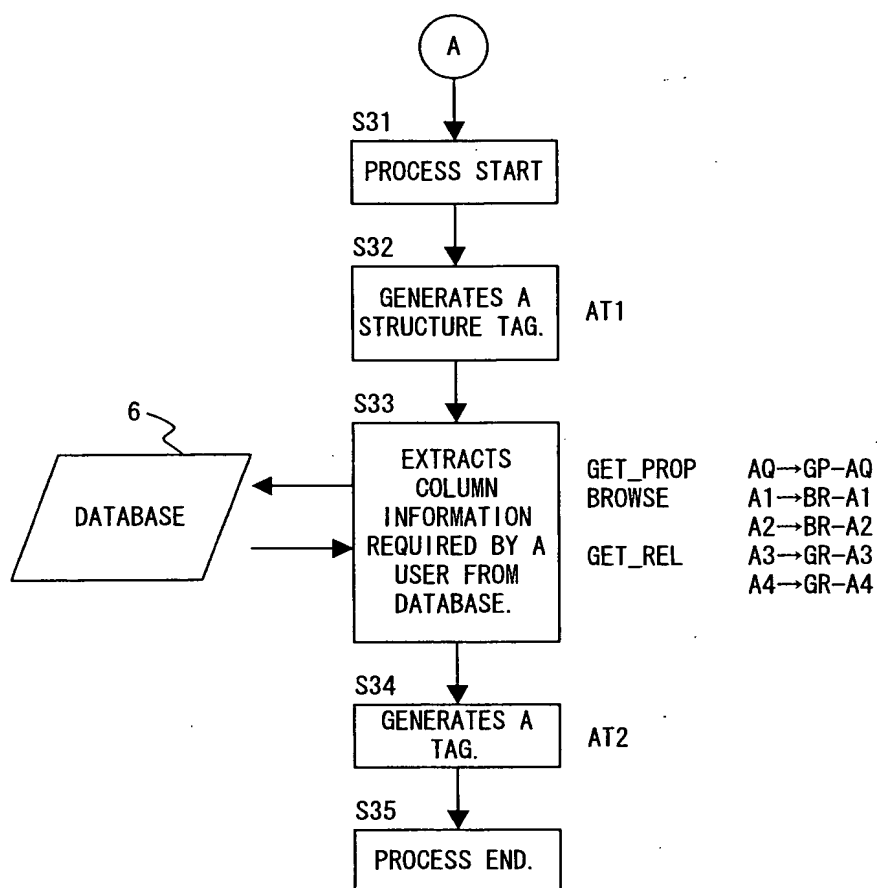
Flowchart showing the operation  
of the present invention



Best Available Copy

[Fig. 3]

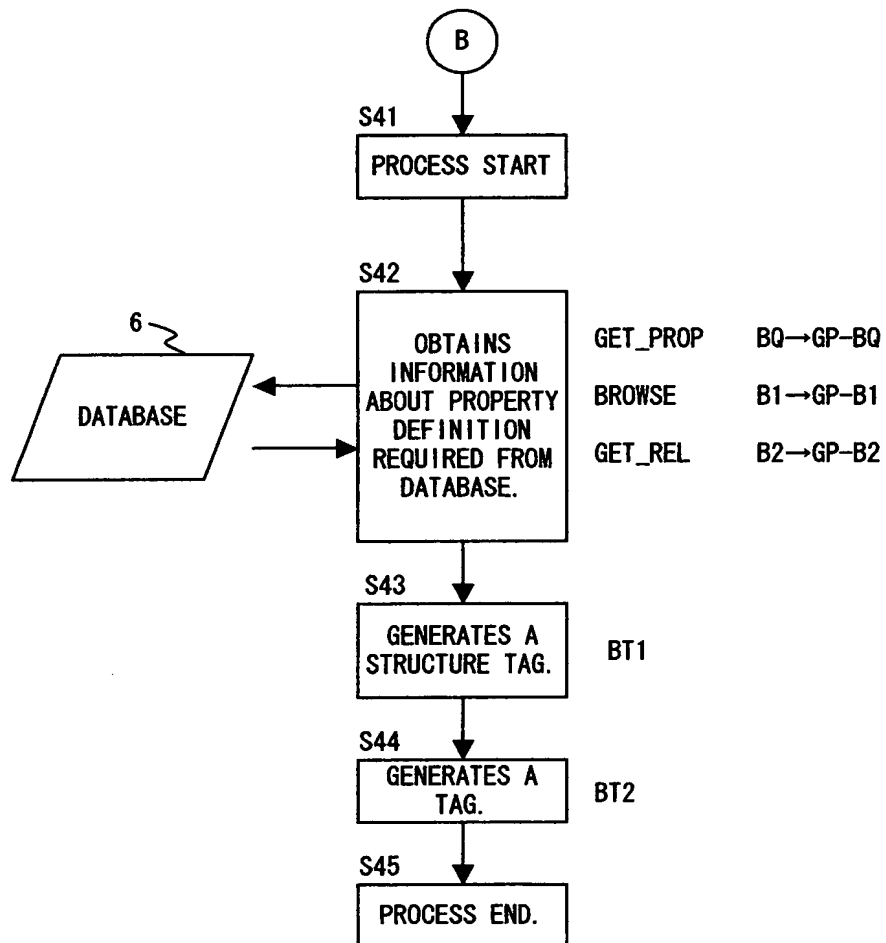
Flowchart showing the detailed operation  
of the present invention  
(a process for inserting property)



Best Available Copy

[Fig. 4]

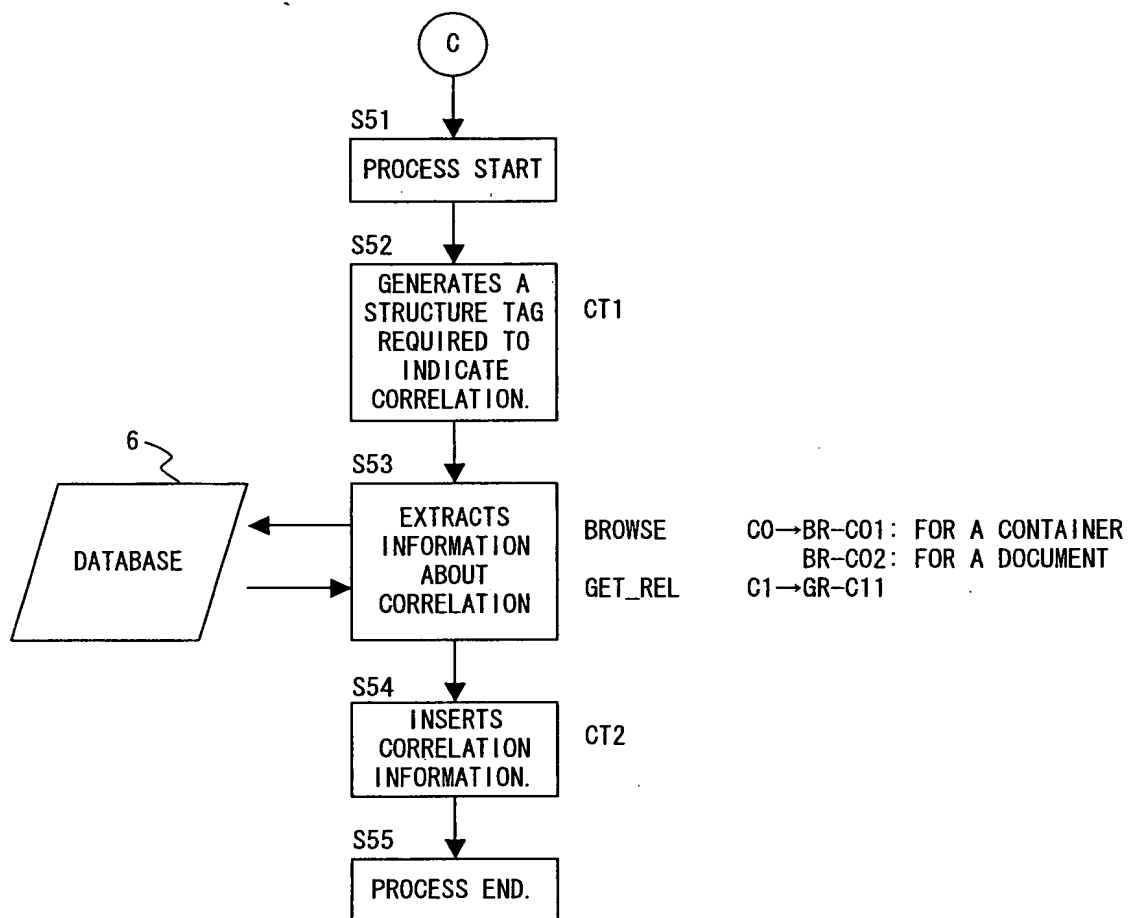
Flowchart showing the detailed operation  
of the present invention  
(information about property)



Best Available Copy

[Fig. 5]

Flowchart showing the detailed operation  
of the present invention  
(a method for inserting correlation)



Best Available Copy

[Fig. 6]

Examples of the requests of the present invention

EXAMPLE 1: REQ=GET\_PROP&REQ\_TYPE=DOCUMENT&DOC\_ID=286

EXAMPLE 2: REQ=BROWSE&MY\_CONT\_ID=22

EXAMPLE 3: REQ=GET\_REL&TYPE=DESTINATION&DOC\_ID=100

**Best Available Copy**

[Fig. 7]

Example of the database used in the present invention

(a) DOCUMENT						
DOCUMENT ID	NAME	WRITER	WRITTEN DAY AND TIME			DOCUMENT SIZE
GR-A4	19 RUBY.doc	HIROBUMI JINNAI	2' 40' '08' ''	p.m.	DECEMBER 21, 1999	300024
	22 asahi.htm	YOSHIYUKI ITO	8' 26' '33' ''	p.m.	JANUARY 14, 2000	637
	24 EUC.htm	HIROBUMI JINNAI	8' 27' '49' ''	p.m.	JANUARY 14, 2000	580
GR-A3	100 exe1500k.xls	HIROBUMI JINNAI	6' 51' '20' ''	a.m.	JANUARY 16, 2000	2347
	286 DOCUMENT A	YOSHIYUKI ITO	2' 51' '00' ''	p.m.	JANUARY 17, 2000	63
	287 DOCUMENT B	YOSHIYUKI ITO	2' 54' '13' ''	p.m.	JANUARY 17, 2000	63

(b) CONTAINER	
CONTAINER ID	NAME
22	MY CONTAINER
25	CONTAINER A
30	CONTAINER B

(c) REFERENCE 1 (INDICATING CORRELATION BETWEEN A CONTAINER AND A DOCUMENT)		
CONTAINER ID	DOCUMENTATION ID	
22	286	
22	287	

(d) REFERENCE 2 (INDICATING CORRELATION BETWEEN CONTAINERS)		
PARENT CONTAINER ID	CHILD CONTAINER ID	
22	25	
22	30	

(e) RELATIONSHIP (INDICATING CORRELATION BETWEEN DOCUMENTS)					
PARENT DOCUMENT ID	CHILD DOCUMENT ID	COMMENT	STRENGTH	INCLUSION	
100	19	COMMENTS PROVIDED	0		0
100	22	COMMENTS PROVIDED	0		0
100	24	COMMENTS PROVIDED	0		0

(f) PROPERTYDEF						
OBJECT ID	REPRESENTATIVE NAME	TYPE	READ FLAG	NECESSITY	MAXIMUM VALUE	MINIMUM VALUE
DOCUMENT	1 DOCUMENT ID	0	1	1		
DOCUMENT	2 WRITER	1	1	1	32	
DOCUMENT	3 WRITTEN DAY AND TIME	2	1	1		
DOCUMENT	5 NAME	1	0	1	255	
DOCUMENT	6 ONLINE VERSION SIZE	1	1	1		
CONTAINER	1 NAME	0	0	1	255	
CONTAINER	6 CONTAINER ID	1	1	1		
RELATIONSHIP	1 EXPLANATION	1	1	0		
RELATIONSHIP	2 STRENGTH	0	1	1	255	
RELATIONSHIP	3 (NOT SUPPORTED)	0	1	1		

[Fig. 8]

Example of the XML of the present invention  
(Example 1)

```

REQUEST
REQ=GET_PROP
REQ_TYPE=DOCUMENT
DOC_ID=286

ACTUALLY GENERATED XML IS AS FOLLOWS.
PAY ATTENTION TO *.
CHARACTER STRINGS PUT BETWEEN ***** ARE EXPLANATION.
HERE, IT IS SET THAT ALL DOCUMENT PROPERTIES ARE REQUIRED.
ALTHOUGH THERE IS refDocument TAG BELOW CONTAINER IN THE PATENT, THE EQUIVALENT EXISTS BELOW success
IN THIS REQUEST.

<?xml version="1.0" standalone="no"?>
<marid:version="1.1">
  <activeContainer key="" browseKey="" containerType="">
    <links>
      <link href="REQ=BROWSE" linkType="browse"/>
      <link href="REQ=GET_PROP&REQ_TYPE=CONT&CONT_ID_0=" linkType="property"/>
      <link href="REQ=DELETE&REQ_TYPE=CONT&CONT_ID_0=" linkType="delete"/>
    </links>
    </activeContainer>
    <results result="success">
      <success>
        <refDocuments>
          <refDocument key="286"/>
        </refDocuments>
      </success>
    </results>
    <links>
      <link href="REQ=EXEC_SCH" linkType="search"/>
      <link href="REQ=BROWSE" linkType="browse"/>
      <link href="REQ=GET_CONFIG&REQ_TYPE=BROWSE" linkType="custom:seBrowse"/>
      <link href="REQ=GET_CONFIG&REQ_TYPE=SCH" linkType="custom:seSearch"/>
      <link href="REQ=GET_CONFIG&REQ_TYPE=SCH_RESULT" linkType="custom:seSearchResult"/>
      <link href="REQ=GET_SSCH" linkType="storedSearch"/>
      <link href="REQ=GET_PROP&REQ_TYPE=USER" linkType="property"/>
      <link href="REQ=GET_PROP&REQ_TYPE=SYSTEM" linkType="system"/>
      <link href="REQ=GET_USERGROUP" linkType="userGroupList"/>
      <link href=""&REFRESH=true" linkType="refresh"/>
      <link href="REQ=LOGOUT" linkType="logout"/>
    </links>
    <documents>
      <document key="286">
        <links>
          <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=286" linkType="retrieve"/>
          <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID_0=286" linkType="property"/>
          <link href="REQ=GET_REL&DOC_ID_0=286" linkType="relationship"/>
          <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=286" linkType="lock"/>
          <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID_0=286" linkType="delete"/>
          <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=286" linkType="checkIn"/>
          <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=286" linkType="checkOut"/>
        </links>
        <properties>
          <property object="document" propertyId="1" key="DOCUMENT_1">286</property>
          <property object="document" propertyId="2" key="DOCUMENT_2">YOSHITUKI ITO</property>
          <property object="document" propertyId="3" key="DOCUMENT_3">2'51'00" p.m. JANUARY 17, 2000</property>
          <property object="document" propertyId="5" key="DOCUMENT_5">DOCUMENT A</property>
        </properties>
        <customProperties/>
      </document>
    </documents>
    <propertyDefinition>
      <fixedPropertyDefinition>
        <propertyDef object="document">
          <propertyDef key="DOCUMENT_1" readOnly="true" required="true" object="document" propertyId="1" selected="browse|sch">
            <displayName>DOCUMENT ID</displayName>
            <limit/>
          </propertyDef>
          <propertyDef key="DOCUMENT_2" readOnly="true" required="true" object="document" propertyId="2" selected="browse" type="
            <displayName>WRITER</displayName>
            <limit max="32"/>
          </propertyDef>
          <propertyDef key="DOCUMENT_3" readOnly="true" required="true" object="document" propertyId="3" selected="sch" type="
            <displayName>WRITTEN DAY AND TIME</displayName>
            <limit/>
          </propertyDef>
          <propertyDef key="DOCUMENT_5" readOnly="false" required="true" object="document" propertyId="5" selected="sch" type="
            <displayName>DESCRIPTION</displayName>
            <limit max="255"/>
          </propertyDef>
        </fixedPropertyDefinition>
      </propertyDefinition>
    </marid:version>
  </marid>

```

AT1

AT2

AT1

AT2

GP - A0

GP - B0  
+  
CT2

BT1

AT1

⑤

④

[Fig. 9]

Example of the database used in the present invention  
(Example 1)

(a) DOCUMENT						
DOCUMENT ID	NAME		WRITER	WRITTEN DAY AND TIME		DOCUMENT SIZE
GR-A4	19	RUBY.doc	HIROBUMI JINNAI	2' 40'' 08'''	p.m. DECEMBER 21, 1999	300024
	22	asahi.htm	YOSHIYUKI ITO	8' 26'' 33'''	p.m. JANUARY 14, 2000	637
	24	EUC.htm	HIROBUMI JINNAI	8' 27'' 49'''	p.m. JANUARY 14, 2000	580
GR-A3	100	exe1500k.xls	HIROBUMI JINNAI	6' 51'' 20'''	a.m. JANUARY 16, 2000	2347
	286	DOCUMENT A	YOSHIYUKI ITO	2' 51'' 00'''	p.m. JANUARY 17, 2000	63
	287	DOCUMENT B	YOSHIYUKI ITO	2' 54'' 13'''	p.m. JANUARY 17, 2000	63

(b) CONTAINER

CONTAINER ID	NAME
22	MY CONTAINER
25	CONTAINER A
30	CONTAINER B

(c) REFERENCE 1 (INDICATING CORRELATION BETWEEN A CONTAINER AND A DOCUMENT)

CONTAINER ID	DOCUMENTATION ID
22	286
22	287

(d) REFERENCE 2 (INDICATING CORRELATION BETWEEN CONTAINERS)

PARENT CONTAINER ID	CHILD CONTAINER ID
22	25
22	30

(e) RELATIONSHIP (INDICATING CORRELATION BETWEEN DOCUMENTS)

PARENT DOCUMENT ID	CHILD DOCUMENT ID	COMMENT	STRENGTH	INCLUSION
100	19	COMMENTS PROVIDED	0	0
100	22	COMMENTS PROVIDED	0	0
100	24	COMMENTS PROVIDED	0	0

(f) PROPERTYDEF

OBJECT ID	NAME	TYPE	READ FLAG	NECESSITY	MAXIMUM VALUE	MINIMUM VALUE
DOCUMENT	1 DOCUMENT ID	0	1	1	32	
DOCUMENT	2 WRITER	1	1	1		
DOCUMENT	3 WRITTEN DAY AND TIME	2	1	1		
DOCUMENT	5 NAME	1	0	1	255	
DOCUMENT	6 ONLINE VERSION SIZE	1	1	1	255	
CONTAINER	1 NAME	0	0	1		
CONTAINER	6 CONTAINER ID	1	1	1		
RELATIONSHIP	1 EXPLANATION	1	1	0	255	
RELATIONSHIP	2 STRENGTH	0	1	1		
RELATIONSHIP	3 (NOT SUPPORTED)	0	1	1		

GP-B0

Best Available Copy



[Fig. 10]

Example of the XML of the present invention  
(Example 2)

```

REQUEST
REQ=BROWSE
MY_CONT_ID=22

ACTUALLY GENERATED XML IS AS FOLLOWS.
CHARACTER STRINGS PUT BETWEEN ***** ARE EXPLANATION.
HERE, IT IS SET THAT DOCUMENT PROPERTIES DOCUMENT_1, DOCUMENT_2, DOCUMENT_6, CONTAINER PROPERTIES
CONTAINER_1 AND CONTAINER_6 ARE REQUIRED.

<?xml version="1.0" standalone="no"?>
<caridlo version="1.1">
  <activeContainer key="22" browseKey="22" containerType="My">
    <links>
      <link href="REQ=BROWSE&MY_CONT_ID=22" linkType="browse"/>
      <link href="REQ=GET_PROP&REQ_TYPE=CONT&CONT_ID_0=22" linkType="property"/>
      <link href="REQ=DELETE&REQ_TYPE=CONT&CONT_ID_0=22" linkType="delete"/>
    </links>
  </activeContainer>
  <results results="success" expire="true">
    <success>
      <refContainers>
        <refContainer key="22"/>
      </refContainers>
    </success>
  </results>
  <links>
    <link href="REQ=EXEC_SCH" linkType="search"/>
    <link href="REQ=BROWSE" linkType="browse"/>
    <link href="REQ=GET_CONF&REQ_TYPE=BROWSE" linkType="customizeBrowse"/>
    <link href="REQ=GET_CONF&REQ_TYPE=SCH" linkType="customizeSearch"/>
    <link href="REQ=GET_CONF&REQ_TYPE=SCH_RESULT" linkType="customizeSearchResult"/>
    <link href="REQ=GET_SSCH" linkType="storedSearch"/>
    <link href="REQ=GET_PROP&REQ_TYPE=USER" linkType="property"/>
    <link href="REQ=GET_PROP&REQ_TYPE=SYSTEM" linkType="system"/>
    <link href="REQ=GET_USERGROUP" linkType="userGroupList"/>
    <link href="REQ=REFRESH=true" linkType="refresh"/>
    <link href="REQ=LOGOUT" linkType="logout"/>
  </links>
  <propertyDefinition>
    <fixedPropertyDefinition>
      <propertyDef object="document">
        <propertyDef key="DOCUMENT_1" readOnly="true" required="true" object="document" propertyId="1" selected="browse">
          <displayName>DOCUMENT ID</displayName>
        </propertyDef>
        <propertyDef key="DOCUMENT_2" readOnly="true" required="true" object="document" propertyId="2" selected="browse">
          <displayName>WRITER</displayName>
        </propertyDef>
        <propertyDef key="DOCUMENT_6" readOnly="true" required="true" object="document" propertyId="6" selected="browse">
          <displayName>ONLINE VERSION SIZE</displayName>
        </propertyDef>
      </fixedPropertyDefinition>
      <propertyDef object="container">
        <propertyDef key="CONTAINER_1" readOnly="false" required="true" object="container" propertyId="1" selected="" type="BR-B1+BT2">
          <displayName>NAME</displayName>
        </propertyDef>
        <propertyDef key="CONTAINER_6" readOnly="true" required="true" object="container" propertyId="6" selected="" type="BR-B1+BT2">
          <displayName>CONTAINER ID</displayName>
        </propertyDef>
      </propertyDef>
    </fixedPropertyDefinition>
  </propertyDefinition>
  <document>
    <document key="286">
      <links>
        <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=286" linkType="retrieve"/>
        <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID_0=286" linkType="property"/>
        <link href="REQ=GET_REL&DOC_ID_0=286" linkType="relationship"/>
        <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=286" linkType="lock"/>
        <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID_0=286" linkType="delete"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=286" linkType="checkIn"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=286" linkType="checkOut"/>
      </links>
      <properties>
        <property object="document" propertyId="1" key="DOCUMENT_1">286</property>
        <property object="document" propertyId="2" key="DOCUMENT_2">YOSHITAKI ITO</property>
        <property object="document" propertyId="6" key="DOCUMENT_6">63</property>
      </properties>
    </document>
    <document key="287">
      <links>
        <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=287" linkType="retrieve"/>
        <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID_0=287" linkType="property"/>
        <link href="REQ=GET_REL&DOC_ID_0=287" linkType="relationship"/>
        <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=287" linkType="lock"/>
        <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID_0=287" linkType="delete"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=287" linkType="checkIn"/>
        <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=287" linkType="checkOut"/>
      </links>
      <properties>
        <property object="document" propertyId="1" key="DOCUMENT_1">287</property>

```

[Fig. 11]

Example of the XML of the present invention  
(Example 2)

```

    <property object="document" propertyId="2" key="DOCUMENT_2">YOSHIYUKI ITO</property>
    <property object="document" propertyId="6" key="DOCUMENT_6">63</property>
  </properties>
</document>
AT1 </documents>
    <containers>
      <container key="22" browseKey="22" active="true" opened="true" containerType="My">
        <links>
          <link href="REQ=BROWSE&MY_CONT_ID=22" linkType="browse"/>
          <link href="REQ=GET_PROP&REQ_TYPE=CONT&CONT_ID_0=22" linkType="property"/>
          <link href="REQ=DELETE&REQ_TYPE=CONT&CONT_ID_0=22" linkType="delete"/>
        </links>
        <properties>
          <property object="container" propertyId="1" key="CONTAINER_1">MY CONTAINER</property>
          <property object="container" propertyId="6" key="CONTAINER_6">22</property>
        </properties>
      </container>
      <container key="25" browseKey="22-25" active="false" opened="false" containerType="My">
        <links>
          <link href="REQ=BROWSE&MY_CONT_ID=22-25" linkType="browse"/>
          <link href="REQ=GET_PROP&REQ_TYPE=CONT&CONT_ID_0=25" linkType="property"/>
          <link href="REQ=DELETE&REQ_TYPE=CONT&CONT_ID_0=25" linkType="delete"/>
        </links>
        <properties>
          <property object="container" propertyId="1" key="CONTAINER_1">CONTAINER A</property>
          <property object="container" propertyId="6" key="CONTAINER_6">25</property>
        </properties>
      </container>
      <container key="30" browseKey="22-30" active="false" opened="false" containerType="My">
        <links>
          <link href="REQ=BROWSE&MY_CONT_ID=22-30" linkType="browse"/>
          <link href="REQ=GET_PROP&REQ_TYPE=CONT&CONT_ID_0=30" linkType="property"/>
          <link href="REQ=DELETE&REQ_TYPE=CONT&CONT_ID_0=30" linkType="delete"/>
        </links>
        <properties>
          <property object="container" propertyId="1" key="CONTAINER_1">CONTAINER B</property>
          <property object="container" propertyId="6" key="CONTAINER_6">30</property>
        </properties>
      </container>
    </containers>
    <refDocuments>
      <refDocument key="286"/>
      <refDocument key="287"/>
    </refDocuments>
  </documents>
AT1 </meridio>

```

BR-A2

BR-A1

BR-C01

BR-C01

BR-A2

BR-C02

Best Available Copy

[Fig. 12]

Example of the database used in the present invention  
(Example 2)

(a) DOCUMENT

DOCUMENT ID	NAME	WRITER	WRITTEN DAY AND TIME	DOCUMENT SIZE
GR-A4	19 RUBY.doc	HIROBUMI JINNAI	2'40''08'' p.m. DECEMBER 21, 1999	300024
	22 asahi.htm	YOSHIYUKI ITO	8'26''33'' p.m. JANUARY 14, 2000	637
	24 EUC.htm	HIROBUMI JINNAI	8'27''49'' p.m. JANUARY 14, 2000	580
GR-A3	100 exe1500k.xls	HIROBUMI JINNAI	6'51''20'' a.m. JANUARY 16, 2000	2347
	286 DOCUMENT A	YOSHIYUKI ITO	2'51''00'' p.m. JANUARY 17, 2000	63
	287 DOCUMENT B	YOSHIYUKI ITO	2'54''13'' p.m. JANUARY 17, 2000	63

(b) CONTAINER

CONTAINER ID	NAME
22	MY CONTAINER
25	CONTAINER A
30	CONTAINER B

BR-A2

(c) REFERENCE 1 (INDICATING CORRELATION BETWEEN A CONTAINER AND A DOCUMENT)

CONTAINER ID	DOCUMENTATION ID	
22	286	BR-CO (CORRELATION)
22	287	BR-CO (CORRELATION)

(d) REFERENCE 2 (INDICATING CORRELATION BETWEEN CONTAINERS)

PARENT CONTAINER ID	CHILD CONTAINER ID	
22	25	BR-CO (CORRELATION)
22	30	BR-CO (CORRELATION)

(e) RELATIONSHIP (INDICATING CORRELATION BETWEEN DOCUMENTS)

PARENT DOCUMENT ID	CHILD DOCUMENT ID	COMMENT	STRENGTH	INCLUSION
100	19	COMMENTS PROVIDED	0	0
100	22	COMMENTS PROVIDED	0	0
100	24	COMMENTS PROVIDED	0	0

(f) PROPERTYDEF

OBJECT ID	NAME	TYPE	READ FLAG	NECESSITY	MAXIMUM VALUE	MINIMUM VALUE
DOCUMENT	1 DOCUMENT ID	0	1	1	32	
DOCUMENT	2 WRITER	1	1	1		
DOCUMENT	3 WRITTEN DAY AND TIME	2	1	1		
DOCUMENT	5 NAME	1	0	1	255	
DOCUMENT	6 ONLINE VERSION SIZE	1	1	1		
CONTAINER	1 NAME	0	0	1		
CONTAINER	6 CONTAINER ID	1	1	1	255	
RELATIONSHIP	1 EXPLANATION	1	1	0		
RELATIONSHIP	2 STRENGTH	0	1	1		
RELATIONSHIP	3 (NOT SUPPORTED)	0	1	1		

GP-B1

Best Available Copy

[Fig. 13]

Example of the XML of the present invention  
(Example 3)

REQUEST  
REQ=GET\_REL  
TYPE=DESTINATION  
DOC\_ID=100

ACTUALLY GENERATED XML IS AS FOLLOWS.

IT IS SET HERE THAT DOCUMENT PROPERTIES DOCUMENT\_1 AND DOCUMENT\_5 ARE REQUIRED.

```

<?xml version="1.0" standalone="no"?>
<meridia version="1.1">
  <activeContainer key="" browseKeys="" containerType="">
    <links>
      <link href="REQ-BROWSE" linkType="browse"/>
      <link href="REQ-GET_PROP&REQ_TYPE=CONT&DOC_ID_0=" linkType="property"/>
      <link href="REQ-DELETE&REQ_TYPE=CONT&DOC_ID_0=" linkType="delete"/>
    </links>
  </activeContainer>
  <results result="success" expires="false">
    <success>
      <refDocuments>
        <refDocument key="00"/>
      </refDocuments>
    </success>
  </results>
  <links>
    <link href="REQ-EXEC_SCH" linkType="search"/>
    <link href="REQ-BROWSE" linkType="browse"/>
    <link href="REQ-GET_CONFIG&REQ_TYPE=BROWSE" linkType="customiseBrowse"/>
    <link href="REQ-GET_CONFIG&REQ_TYPE=SCH" linkType="customiseSearch"/>
    <link href="REQ-GET_CONFIG&REQ_TYPE=SCH_RESULT" linkType="customiseSearchResult"/>
    <link href="REQ-GET_SSH" linkType="storedSearch"/>
    <link href="REQ-GET_PROP&REQ_TYPE=USER" linkType="property"/>
    <link href="REQ-GET_PROP&REQ_TYPE=SYSTEM" linkType="system"/>
    <link href="REQ-GET_USERGROUP" linkType="userGroupList"/>
    <link href="REQ-REFRESH&REQ_REFRESH=TRUE" linkType="refresh"/>
    <link href="REQ-LOGOUT" linkType="logout"/>
  </links>
  <propertyDefinition>
    <fixedPropertyDefinition>
      <propertyDef object="document">
        <propertyDef key="DOCUMENT_1" readOnly="true" required="true" object="document" propertyId="1" selected="browse|sch">
          <displayName>DOCUMENT ID</displayName>
        </propertyDef>
        <propertyDef key="DOCUMENT_5" readOnly="false" required="true" object="document" propertyId="5" selected="sch" type="document">
          <displayName>DESCRIPTION</displayName>
        </propertyDef>
      </propertyDef>
      <propertyDef object="relationship">
        <propertyDef key="RELATIONSHIP_1" readOnly="true" required="false" object="relationship" propertyId="1" selected="" type="relationship">
          <displayName>EXPLANATION</displayName>
        </propertyDef>
        <propertyDef key="RELATIONSHIP_2" readOnly="true" required="true" object="relationship" propertyId="2" selected="" type="relationship">
          <displayName>STRENGTH</displayName>
        </propertyDef>
        <propertyDef key="RELATIONSHIP_3" readOnly="true" required="true" object="relationship" propertyId="3" selected="" type="relationship">
          <displayName>(NOT SUPPORTED)</displayName>
        </propertyDef>
      </propertyDef>
    </fixedPropertyDefinition>
  </propertyDefinition>
  <document key="10">
    <links>
      <link href="REQ-GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=19" linkType="retrieve"/>
      <link href="REQ-GET_PROP&REQ_TYPE=DOC&DOC_ID_0=19" linkType="property"/>
      <link href="REQ-GET_REL&DOC_ID_0=19" linkType="relationship"/>
      <link href="REQ-GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=19" linkType="lock"/>
      <link href="REQ-DELETE&REQ_TYPE=DOC&DOC_ID_0=19" linkType="delete"/>
      <link href="REQ-UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=19" linkType="checkIn"/>
      <link href="REQ-UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=19" linkType="checkOut"/>
    </links>
    <properties>
      <property object="document" propertyId="1" key="DOCUMENT_1">19</property>
      <property object="document" propertyId="5" key="DOCUMENT_5">REL.doc</property>
    </properties>
  </document>
  <document key="22">
    <links>
      <link href="REQ-GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=22" linkType="retrieve"/>
      <link href="REQ-GET_PROP&REQ_TYPE=DOC&DOC_ID_0=22" linkType="property"/>
      <link href="REQ-GET_REL&DOC_ID_0=22" linkType="relationship"/>
      <link href="REQ-GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=22" linkType="lock"/>
      <link href="REQ-DELETE&REQ_TYPE=DOC&DOC_ID_0=22" linkType="delete"/>
      <link href="REQ-UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=22" linkType="checkIn"/>
      <link href="REQ-UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=22" linkType="checkOut"/>
    </links>
    <properties>
      <property object="document" propertyId="1" key="DOCUMENT_1">22</property>
      <property object="document" propertyId="5" key="DOCUMENT_5">asahi.htm</property>
    </properties>
  </document>
</meridia>

```

Best Available Copy

[Fig. 14]

Example of the XML of the present invention  
(Example 3)

```

    </properties>
  </document>
  <document key="24">
    <links>
      <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=24" linkType="retrieve"/>
      <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID_0=24" linkType="property"/>
      <link href="REQ=GET_REL&DOC_ID_0=24" linkType="relationship"/>
      <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=24" linkType="lock"/>
      <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID_0=24" linkType="delete"/>
      <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=24" linkType="checkin"/>
      <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=24" linkType="checkout"/>
    </links>
    <properties>
      <property object="document" propertyId="1" key="DOCUMENT_1">24</property>
      <property object="document" propertyId="5" key="DOCUMENT_5">EUC.htm</property>
    </properties>
  </document>
  <document key="100">
    <links>
      <link href="REQ=GET_CONTENT&REQ_TYPE=VER&DOC_ID_0=100" linkType="retrieve"/>
      <link href="REQ=GET_PROP&REQ_TYPE=DOC&DOC_ID_0=100" linkType="property"/>
      <link href="REQ=GET_REL&DOC_ID_0=100" linkType="relationship"/>
      <link href="REQ=GET_PROP&REQ_TYPE=LOCK&DOC_ID_0=100" linkType="lock"/>
      <link href="REQ=DELETE&REQ_TYPE=DOC&DOC_ID_0=100" linkType="delete"/>
      <link href="REQ=UPDATE&REQ_TYPE=CHECK_IN&DOC_ID_0=100" linkType="checkin"/>
      <link href="REQ=UPDATE&REQ_TYPE=CHECK_OUT&DOC_ID_0=100" linkType="checkout"/>
    </links>
    <properties>
      <property object="document" propertyId="1" key="DOCUMENT_1">100</property>
      <property object="document" propertyId="5" key="DOCUMENT_5">exos1500k.xls</property>
    </properties>
  </document>
  <destinationDocuments>
    <destinationDocument>
      <refDocument key="19"/>
      <relationship destinationKey="19">
        <links>
          <link href="REQ=DELETE&REQ_TYPE=REL&DOC_ID_0=100&DESTINATION_DOC_ID=19" linkType="delete"/>
        </links>
        <properties>
          <property object="relationship" propertyId="1" key="RELATIONSHIP_1">COMMENTS PROVIDED</property>
          <property object="relationship" propertyId="2" key="RELATIONSHIP_2">0</property>
          <property object="relationship" propertyId="3" key="RELATIONSHIP_3">0</property>
        </properties>
      </relationship>
    </destinationDocument>
    <destinationDocument>
      <refDocument key="22"/>
      <relationship destinationKey="22">
        <links>
          <link href="REQ=DELETE&REQ_TYPE=REL&DOC_ID_0=100&DESTINATION_DOC_ID=22" linkType="delete"/>
        </links>
        <properties>
          <property object="relationship" propertyId="1" key="RELATIONSHIP_1">COMMENTS PROVIDED</property>
          <property object="relationship" propertyId="2" key="RELATIONSHIP_2">0</property>
          <property object="relationship" propertyId="3" key="RELATIONSHIP_3">0</property>
        </properties>
      </relationship>
    </destinationDocument>
    <destinationDocument>
      <refDocument key="24"/>
      <relationship destinationKey="24">
        <links>
          <link href="REQ=DELETE&REQ_TYPE=REL&DOC_ID_0=100&DESTINATION_DOC_ID=24" linkType="delete"/>
        </links>
        <properties>
          <property object="relationship" propertyId="1" key="RELATIONSHIP_1">COMMENTS PROVIDED</property>
          <property object="relationship" propertyId="2" key="RELATIONSHIP_2">0</property>
          <property object="relationship" propertyId="3" key="RELATIONSHIP_3">0</property>
        </properties>
      </relationship>
    </destinationDocument>
  </destinationDocuments>
</documents>
</veridic>

```

AT2

GR-A4

CT1E

GR-A3

GR-C11

CT1E

AT2E

AT1E

[Fig. 15]

Example of the database used in the present invention  
(Example 3)

(a) DOCUMENT						
DOCUMENT ID	NAME	WRITER	WRITTEN DAY AND TIME			DOCUMENT SIZE
GR-A4	19 RUBY.doc	HIROBUMI JINNAI	2' 40'' 08''	p. m.	DECEMBER 21, 1999	300024
	22 asahi.htm	YOSHIYUKI ITO	8' 26'' 33''	p. m.	JANUARY 14, 2000	637
	24 FUG.htm	HIROBUMI JINNAI	8' 27'' 49''	p. m.	JANUARY 14, 2000	580
GR-A3	100 exe1500k.xls	HIROBUMI JINNAI	6' 51'' 20''	a. m.	JANUARY 16, 2000	2347
	286 DOCUMENT A	YOSHIYUKI ITO	2' 51'' 00''	p. m.	JANUARY 17, 2000	63
	287 DOCUMENT B	YOSHIYUKI ITO	2' 54'' 13''	p. m.	JANUARY 17, 2000	63

(b) CONTAINER		
CONTAINER ID	NAME	
22	MY CONTAINER	
25	CONTAINER A	
30	CONTAINER B	

(c) REFERENCE 1 (INDICATING CORRELATION BETWEEN A CONTAINER AND A DOCUMENT)		
CONTAINER ID	DOCUMENTATION ID	
22	286	
22	287	

(d) REFERENCE 2 (INDICATING CORRELATION BETWEEN CONTAINERS)		
PARENT CONTAINER ID	CHILD CONTAINER ID	
22	25	
22	30	

(e) RELATIONSHIP (INDICATING CORRELATION BETWEEN DOCUMENTS)					
PARENT DOCUMENT ID	CHILD DOCUMENT ID	COMMENT	STRENGTH	INCLUSION	
100	19	COMMENTS PROVIDED	0	0	GR-C1 (CORRELATION)
100	22	COMMENTS PROVIDED	0	0	
100	24	COMMENTS PROVIDED	0	0	

(f) PROPERTYDEF							
OBJECT ID	NAME	TYPE	READ FLAG	NECESSITY	MAXIMUM VALUE	MINIMUM VALUE	
DOCUMENT	1 DOCUMENT ID	0	1	1			
DOCUMENT	2 WRITER	1	1	1	32		
DOCUMENT	3 WRITTEN DAY AND TIME	2	1	1			
DOCUMENT	5 NAME	1	0	1	255		
DOCUMENT	6 ONLINE VERSION SIZE	1	1	1			
CONTAINER	1 NAME	0	0	1	255		
CONTAINER	6 CONTAINER ID	1	1	1			
RELATIONSHIP	1 EXPLANATION	1	1	0			
RELATIONSHIP	2 STRENGTH	0	1	1	255		
RELATIONSHIP	3 (NOT SUPPORTED)	0	1	1			

GP-B2

Best Available Copy